QUESTION SET-5

**QUESTION-1: Flights Collection**

**a) Retrieve all flights whose destination is either "New York" or "London".**

db.Flights.find({ destination: { $in: ["New York", "London"] } })

**b) Find flights that have more than 100 passengers booked.**

db.Flights.find({ "passengers": { $size: { $gt: 100 } } })

**c) Identify flights scheduled to depart on both "Monday" and "Friday".**

db.Flights.find({ schedule: { $all: ["Monday", "Friday"] } })

**d) Sort flights in ascending order of departure time and limit to 5 records.**

db.Flights.find()

.sort({ "details.departureTime": 1 })

.limit(5)

**e) Group flights by destination and count the total flights for each.**

javascript

CopyEdit

db.Flights.aggregate([

{ $group: { \_id: "$destination", totalFlights: { $sum: 1 } } }

])

**QUESTION-2: Passengers Collection**

**a) Retrieve passengers who booked more than 3 flights but do not prefer "Business Class".**

db.Passengers.find({

"flightsBooked": { $size: { $gt: 3 } },

"preferences": { $ne: "Business Class" }

})

**b) Find passengers who booked both "Flight 101" and "Flight 202".**

db.Passengers.find({ flightsBooked: { $all: ["Flight 101", "Flight 202"] } })

**c) Display names and contact details of passengers whose passport number starts with "A123".**

db.Passengers.find(

{ passportNumber: { $regex: "^A123" } },

{ \_id: 0, name: 1, contact: 1 }

)

**d) Create a compound index on passportNumber and contact.email for uniqueness.**

db.Passengers.createIndex(

{ passportNumber: 1, "contact.email": 1 },

{ unique: true }

)

**e) Aggregate to show name, passport number, and total flights booked (filtering out passengers with <2 flights).**

db.Passengers.aggregate([

{ $project: { name: 1, passportNumber: 1, totalFlights: { $size: "$flightsBooked" } } },

{ $match: { totalFlights: { $gte: 2 } } }

])